Parallel Session 6B
Basic, Extended & Advanced Examples

I. Hrivnacova, S. Guatteli

23rd Geant4 Collaboration Meeting,
31 August 2018, Lund

Current status of the Geant4 Advanced Examples	*Susanna Guatelli* 14:00 - 14:15
Hadrontherapy: status, recent updates, and new developments	*G. Petringa* 14:15 - 14:30
B&E Examples WG: Work plan items	*Ivana Hrivnacova* 14:30 - 14:45
VIS, UI clean-up	*John Allison* 14:45 - 14:55
New OpNovice2 example	*Daren Sawkey* 14:55 - 15:00
New extended/physicslists category	*Ivana Hrivnacova* 15:00 - 15:05
New extensibleFactory example	*Robert William Hatcher* 15:05 - 15:10
DICOM2 + Proposal for new demonstration of using statistic tools	*Jonathan Madsen* 15:10 - 15:25

Advanced examples- Summary

Coordinator: S. Guatelli, Deputy Coordinator: F. Romano

Recent developments since last public release:

- **Migration to MultiFunctionalDetector/scoring mesh– remove RO Geometry**
  - IORT_therapy, Medical_linac. Authors: Caccia, Guatelli, Pisciotta, Russo
- **GammaRayTelescope**
  - https://bugzilla-geant4.kek.jp/show_bug.cgi?id=1981: solved
  - Authors: L. Pandola and F. Longo
- **Hadrontherapy**
  - Radiobiological modelling added. Authors: G. Petringa, P. Cirrone, L. Pandola
- **Brachytherapy**
  - Add I-125 source with validation against ref data. Authors: A. Lee, D. Cutajar and S. Guatelli

New proposed example

- Nuclear medicine/PET – missing now
- Authors:
  - M. Safavi & A. Ahmed, Australian Nuclear Science Technology and Organisation
  - A. Chacon and S. Guatelli, University of Wollongong
- Maintenance in Geant4: S. Guatelli
- T. Yamaya, H. Tashima, E. Yoshida, G. Akamatsu, A. Mohammadi, NIRS, supported the study by providing the technical details of the PET scanner

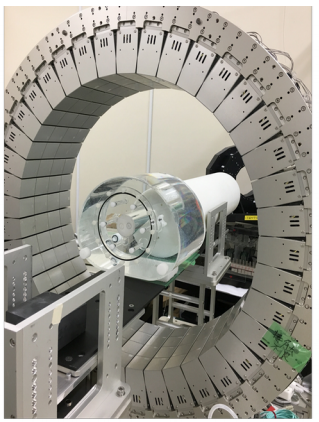
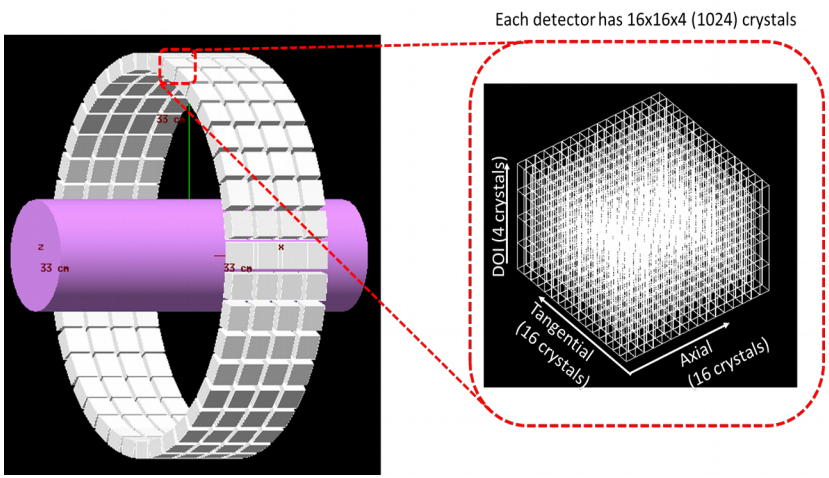
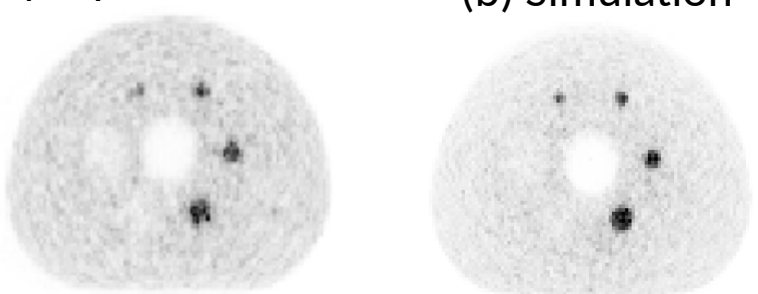


Fig. DOI-enabled whole-body PET scanner (image from Taiga-lab, NIRS, Japan)



(a) Experimental

(b) Simulation



CENTRE FOR MEDICAL RADIATION PHYSICS

Something to think about/discuss

- Wide experimental coverage:
  - HEP (15%), Space science/astrophysics (20%), Medical physics and radiobiology (40%), Detector technologies and others (25%)
- The Advanced Examples may be published
  - In themes (med phys, space science, High Energy Physics)
  - The papers should go through the Geant4 Editorial Board
- We need to promote the Advanced Examples in conferences/workshops. Next ones:
  - Bordeaux Workshop 2018 – oral presentation
  - NSS-MIC 2018 - poster

Extended Examples Common Tasks

I. Hrivnacova

- The list of explicitly defined physics lists and physics builders
  - To be further looked at by Physics Lists WG
- Code review concentrated on UI commands and macros
- Coding guidelines
  - Most of violations in 3 examples: which were not yet updated at all:
  - HepMC/HepMCEx01, HepMC/HepMCEx02 - not yet updated at all
  - medical/DICOM/dicomReader – using a different convention for class data membes (“the” instead of “f”) , can be excluded from checking
- Build Options in DICOM example need to be handled in ctest

Vis and UI in examples

John Allison

Geant4 Associates International Ltd
and
The University of Manchester

Geant4 Collaboration Meeting
Lund August 2018

- Removal of G4VIS_USE and G4_UI flags
- The introduction of Graphical User Interfaces places some new requirements on the design of main()
- Many examples instantiate the UI too late.
- GUIs capture G4cout, so it should be instantiated at the very beginning of main(), otherwise some output will go to std::cout/cerr.

	Old (TestEm1.cc)	New (exampleB1.cc)

#include statements	`#ifdef G4VIS_USE`
	`#include "G4VisExecutive.hh"`
	`#endif`
	
	`#ifdef G4UI_USE`
	`#include "G4UIExecutive.hh"`
	`#endif`
		`#include "G4VisExecutive.hh"`
		`#include "G4UIExecutive.hh"`

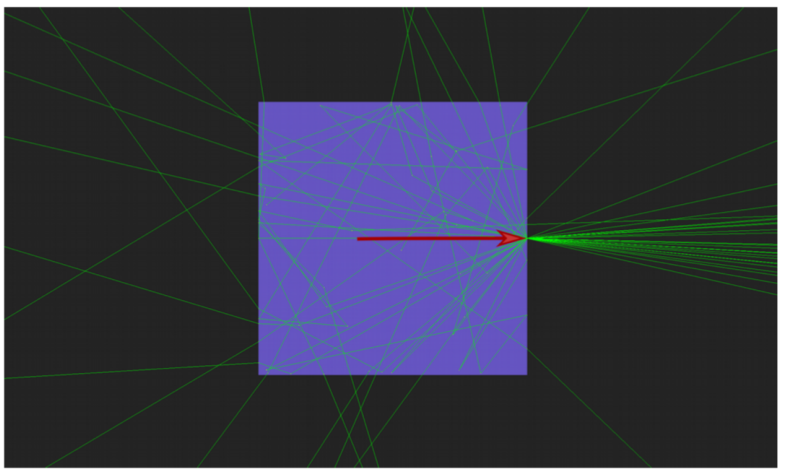
In interactive mode, G4UIExecutive is instantiated straight away so that for graphical user interfaces all G4cout output is captured from the start.
 	
ui will be used as a flag for interactive mode.
	`int main(int argc,char** argv) {`
	`...`
		`int main(int argc,char** argv)`
		`{`
		`G4UIExecutive* ui = 0;`
		`if ( argc == 1 ) {`
		`ui = new G4UIExecutive(argc, argv);`
		`}`
		`...`

The logic here is maintained. Note: it's usual to invoke vis.mac or init_vis.mac in interactive mode before starting the session.	`G4UImanager* UI = G4UImanager::GetUIpointer();`
	`if (argc!=1) // batch mode`
	`{`
	`G4String command = "/control/execute ";`
	`G4String fileName = argv[1];`
	`UI->ApplyCommand(command+fileName);`
	`}`
	
	`else // interactive mode`
	`{`
	`#ifdef G4VIS_USE`
	`G4VisManager* visManager = new G4VisExecutive;`
	`visManager->Initialize();`
	`#endif`
	
	`#ifdef G4UI_USE`
	`G4UIExecutive * ui = new G4UIExecutive(argc,argv);`
	`ui->SessionStart();`
	`delete ui;`
	`#endif`
	
	`#ifdef G4VIS_USE`
	`delete visManager;`
	`#endif`
	`}`
		`G4VisManager* visManager = new G4VisExecutive;`
		`visManager->Initialize();`
		
		`G4UImanager* UImanager = G4UImanager::GetUIpointer();`
		`if ( ! ui ) {`
		`// batch mode`
		`G4String command = "/control/execute ";`
		`G4String fileName = argv[1];`
		`UImanager->ApplyCommand(command+fileName);`
		`}`
		`else {`
		`// interactive mode`
		`UImanager->ApplyCommand("/control/execute init_vis.mac");`
		`ui->SessionStart();`
		`delete ui;`
		`}`
		
		`delete visManager;`

OpNovice2

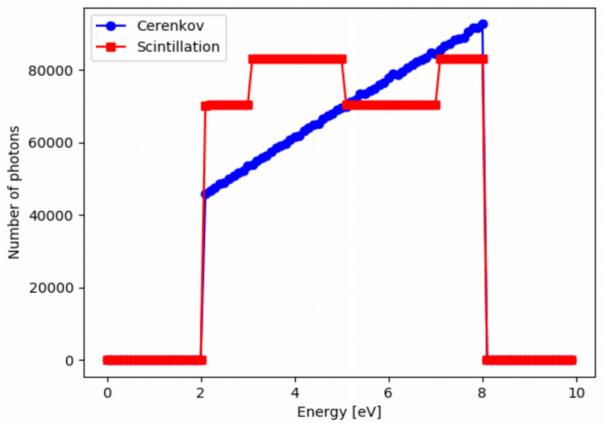
D. Sawkey

Designed to be easy to use



- Material property messenger
- Simple geometry (box in world box)
- Physics constructor including G4OpticalPhysics
- Histogram and table output

Example histogram: photon spectra

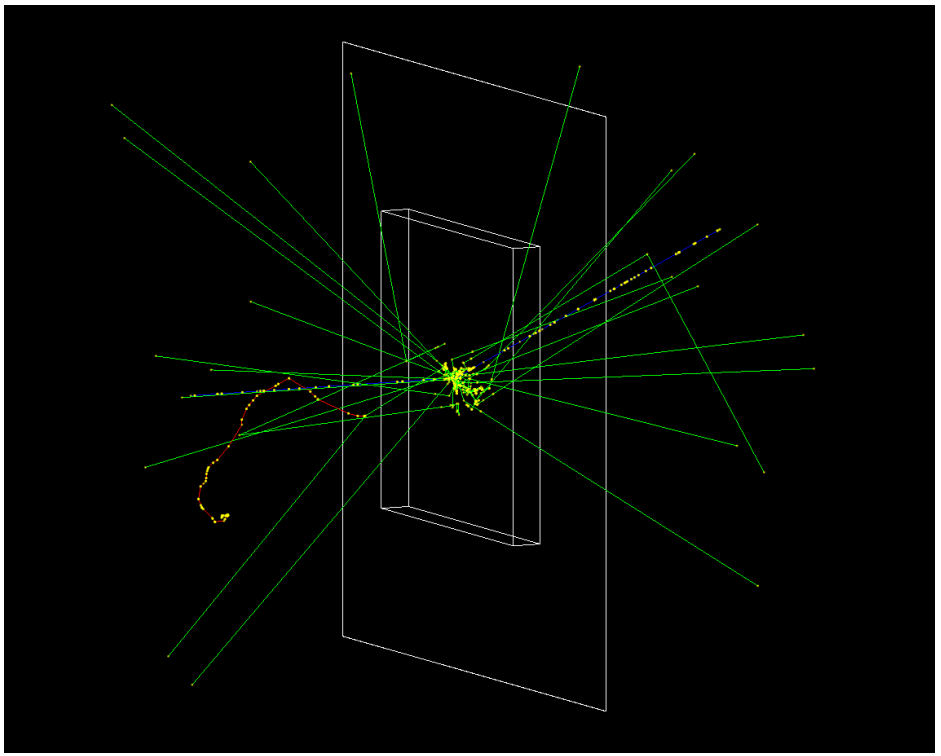


- Follows EM TestEm examples
- Developed internally to answer HyperNews questions
- Previous code with material properties hardwired doesn't scale, hard to share
- Further development foreseen:
  - More histograms, validation and consistency checks of optical physics
  - Material properties messenger in Geant4 source (maybe)

Physicslists Category I. Hrivnacova

- Introduced in the last year release (10.4)
- The purpose is to demonstrate usage of Geant4 reference physics lists and physics builders.
- Three examples:
  - factory, extensibleFactory, genericPL
- The same scenario is used in all three example
  - Implemeneted with use of shared code

Physicslists Examples Scenario



- Geometry: a box of scintillator material (CsI) followed by a thin box of air (screen) which is used to simplify scoring
- The primary generator: G4ParticleGun; default 1 GeV proton
- The screen volume is associated with a sensitive detector, ScreenSD
- The scored quantities are filled in the Screen ntuple, which is defined using G4AnalysisManager and is saved in a Root file

extensibleFactory example

R. Hatcher

Purpose: Demonstrates use of extensible factory

Built upon “factory” example as a base.

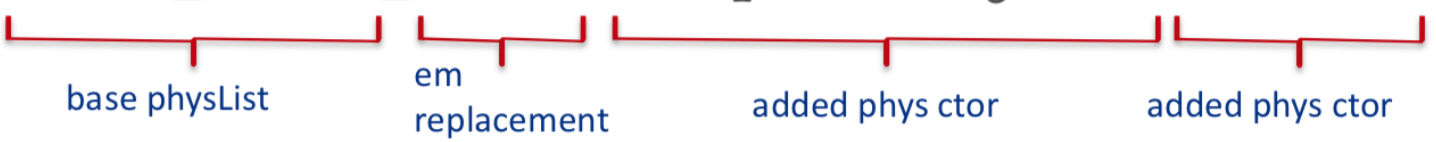
- Only necessary change to switch is:

```
-   #include "G4PhysListFactory.hh"
+   #include "G4PhysListFactoryAlt.hh"
+   using namespace g4alt;
+  // no further changes to the code are required
```

Extending existing physics lists

Out of the box the alternative factory allows all known physics lists to be extended by adding / replacing physics constructors:

```
-p FTFP_BERT_EMX+G4OpticalPhysics+RADIO
```



+ = adds physics constructor via RegisterPhysics()
_ = replaces physics constructor via ReplacePhysics()

Other features

Demonstrate registering application specific physics list "MySpecialPhysList" with the extensible factory:

```
#include "G4PhysListStamper.hh"  // defines macro for factory registration
#include "MySpecialPhysList.hh"
G4_DECLARE_PHYSLIST_FACTORY(MySpecialPhysList);
```

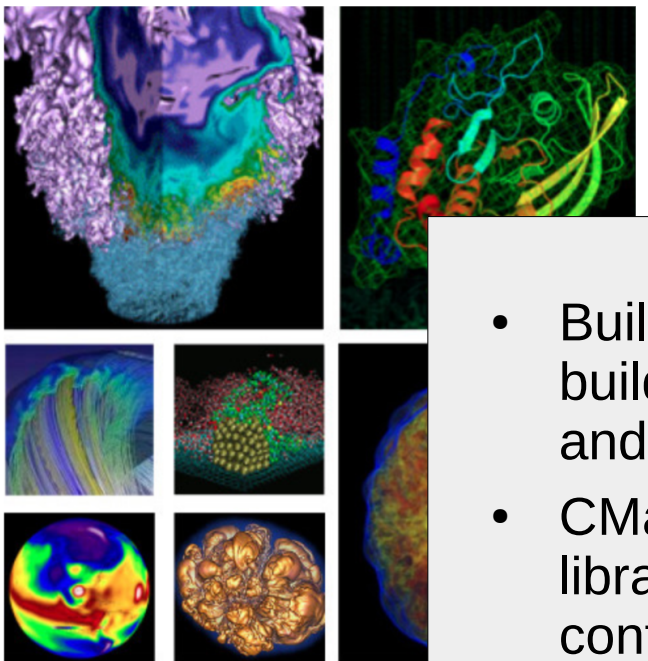
…. and more

DICOM2

J. R. Madsen

DICOM2 Extended Example

Using another example as library and demonstration of hits + statistics



NERSC National Energy Research Scientific Computing Center

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Jonathan R. Madsen

✉ jrmadsen@lbl.gov

National Energy Research Scientific Co

Lawrence Berkeley National Laboratory

August 30, 2018

- Build DICOM/src as library, build DICOM2.cc as exe and link to “DICOM” library
- CMake macro for building a library and a CMake configuration file
- Customized Run, RunAction implementation
- New scoring features for run accumulation scoring

- MCNP always provides statistics by default
- Monte Carlo simulations are statistical simulations
- Many users, very reasonably, want statistics for their answers
- We don't feature statistics in our basic examples even though statistics are a *basic* part of Monte Carlo simulations
- Because we only feature statistics in extended and advanced examples, many users think they have to calculate their own statistics

- Proposal to add simple statistics in some basic example (using G4 classes from globals/HEPNumerics)